



Jiang, H., Fioranelli, F., Yang, S., Romain, O. and Le Kernec, J. (2021) Human Activity Classification Using Radar Signal and RNN Networks. In: IET International Radar Conference 2020, Chongqing City, China, 4-6 Nov 2020, pp. 1595-1599. ISBN 9781839535406.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

<http://eprints.gla.ac.uk/223729/>

Deposited on: 1 October 2020

Enlighten – Research publications by members of the University of Glasgow  
<http://eprints.gla.ac.uk>

# HUMAN ACTIVITY CLASSIFICATION USING RADAR SIGNAL AND RNN NETWORKS

*Haoyang Jiang<sup>1</sup>, Francesco Fioranelli<sup>2</sup>, Shufan Yang<sup>3</sup>, Olivier Romain<sup>4</sup>, Julien Le Kernec<sup>1,3,4</sup>*

*<sup>1</sup>School of Information and Communication, University of Electronic Science and Technology of China, Chengdu, China*

*<sup>2</sup>MS3-Microwave Sensing Signals and Systems, TU Delft, Delft, The Netherlands*

*<sup>3</sup>James Watt School of Engineering, University of Glasgow, Glasgow, UK*

*<sup>4</sup>ETIS – Signal and Information Processing lab, University Cergy-Pontoise, Cergy, France*

*\*Julien.lekernec@glasgow.ac.uk*

**Keywords:** CLASSIFICATION, HUMAN ACTIVITY RECOGNITION, RADAR SIGNAL, LSTM

## 1. Abstract

Radar-based human activities recognition is still an open problem and is a key to detect anomalous behaviour for security and health applications. Deep learning networks such as convolutional neural networks (CNN) have been proposed for such tasks and showed better performance than traditional supervised learning paradigm. However, it is hard to deploy CNN networks to embedded systems due to the limited computational power available. From this point of concern, the use of a recurrent neural network (RNN) is proposed in this paper for human activities classification. We also propose an innovative data argumentation method to train the neural network using a limited number of data. The experiment shows that our network can achieve a mean accuracy of 94.3% in human activity classification.

## 2. Introduction

Action recognition is a research hotspot, which has important applications in pedestrian monitoring, urban facilities security and other public fields. Most of the current research on human action recognition is based on image or video data [1] [2] [3], because, on the one hand, video and image are very intuitive to humans. On the other hand, neural network-based image recognition and segmentation technology have made significant progress.

Due to the ageing trend of the global population and the multimorbidity associated with it, demand for human activity monitoring technology in the private sector is increasing, such as fall detection and patient monitoring. However, because of the concern about privacy security, image-based technology cannot be widely applied in this field. In this case, radar-based activities recognition technology is proposed because it does not need to record ordinary optical images, reducing privacy risks [4].

In the early stage, researches mainly focus on traditional supervised learning paradigms, including but not limited to using time-varying signatures extracted from a spectrogram to classify various human activities [5], empirical mode decomposition method to extract spectral features, and then use support vector machine to identify human motions [6], improved principal component analysis and linear discriminant analysis can also be used to extract features [7], linear predictive code was applied for real-time recognition to reduce the computational cost [8].

These studies have achieved high accuracy results in this application, but these methods require handcrafted features to perform classification or recognition. Such dependence on the domain knowledge limits the application of these methods. Therefore, deep learning-based methods were then introduced into this field.

Deep learning technologies have shown powerful abilities to process radar signals [9] [10]. Previous works include using CNN to detect human and classify human activities [11], and using Deep Auto-Encoders based method to detect human fall [12]. CNN or neural networks with similar architectures treat the input spectrogram as an image and perform detection or classification on image discarding the temporal information. However, spectrograms are not just images but represent the evolution of the target's mean motion and micro-motions as a function of time. These have physical meaning linked to the kinematics of human motion. In this paper, we propose the use of RNN in this task and the spectrogram is regarded as a sequential signal to exploit this temporal information using different kinds of RNN.

We also propose a novel data argumentation method on radar signals. To the best of our knowledge, the database we used in this paper is one of the biggest in radar for human activity recognition, but it is still small for training a neural

network. In this paper, we demonstrate that using a small dataset to train a network will cause overfitting and degrade the performance of the network. So instead of using fixed sample sizes, we propose to use random sizes and parts of the samples to train the networks and therefore augment the size of the dataset. We will demonstrate that this method can speed up the training process and significantly reduce the overfitting phenomenon.

### 3. Radar Data and Classification Networks

#### 3.1. Radar data

The recorded data were obtained using an Ancortek 580AD Frequency Modulated Continuous Wave (FMCW) radar system. The transmitted signal has an instantaneous bandwidth of 400MHz centred at 5.8GHz, and a pulse repetition period of 1ms. The output power of the transmitting amplifier is ~18dBm and is connected to a Yagi antenna with a gain of ~17dB. The receiver antenna is identical and the dechirped signal, i.e., the beat frequency is sampled at 128kHz with in-phase and in-quadrature channels (IQ).

The entire database contains 1754 radar signatures from 72 participants aged 21 to 98 years old. The six different types of daily activities are shown in Figure 1. The class 'Fall' (6) has fewer samples than the other five classes because the elders could not be asked to perform falling due to the research ethics and security concerns.

Table 1 List of activities

No.	Activity Description
(1)	Walking back and forth
(2)	Sitting down on a chair
(3)	Standing up from a chair
(4)	Picking up an object from the ground
(5)	Drinking water from a glass
(6)	Fall

#### 3.2. Classification Networks

There are two reasons why we chose different deep learning structure from previous works. One reason is that RNN structures can handle input data of any length. In this database, samples with different labels have different length, as shown in Figure 1. For example, the samples of 'Walking back and forth' class (1) last for 10 seconds or 20 seconds, but the samples of 'standing up' class (3) only last 5 seconds or less, which means the data input to the network need to have different length. Some researches [13] improved classic CNN structures so that they can be adapted to handle arbitrary size input, but this kind of structures are used for images with big sizes, so it is hard to apply in this case.

Secondly, RNNs are employed to reduce the computational load of the neural network for deployment on embedded systems with limited computational power available. Using CNN, each time window of the spectrogram is fed to the network, because CNN-based structures do not have memory

units, the network will regard them as independent inputs and compute the output individually. When monitoring with small time intervals between samples, two sequential inputs for a CNN, have much overlap. This means a lot of calculations are performed for each time step and is therefore wasteful. However, when using RNN structures, since it has memory units, small timestep monitoring can be implemented by feeding a small piece of the spectrogram into the RNN each time, without sacrificing the computational load.

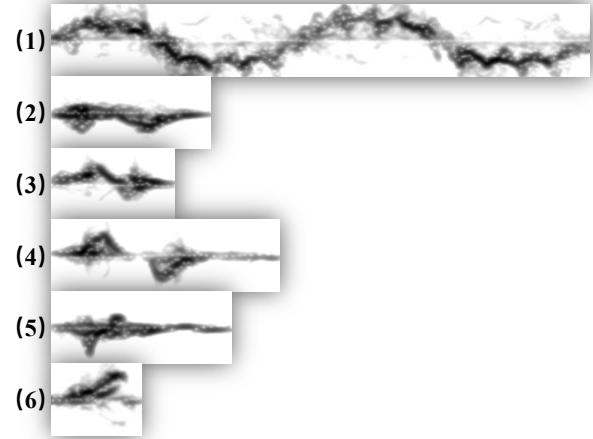


Figure 1 The duration of the signals is variable. The samples from top to bottom are: (1) Walking back and forth, 10.0s; (2) Sitting down on a chair, 2.9s; (3) Standing up from a chair, 2.2s; (4) Picking up an object from the ground, 4.2s; (5) Drinking water from a glass, 3.3s; (6) Fall, 1.7s;

Figure 2 and Figure 3 demonstrate the difference between CNN and RNN, respectively. Every second, a 4s-long segment of the spectrogram is fed to the CNN network to perform a prediction. However, this monitoring rate can also be achieved by feeding spectrograms with 1s-long segments to the RNN network every second. Assuming CNN and RNN algorithms take the same computational load to process inputs with the same duration, the computational load of the CNN structure is four times higher than the RNN. Note that the computational load of CNN can be reduced by using smaller input durations, but when the input becomes smaller, the amount of information available will also decrease, which will cause a degradation of the CNN network performance.

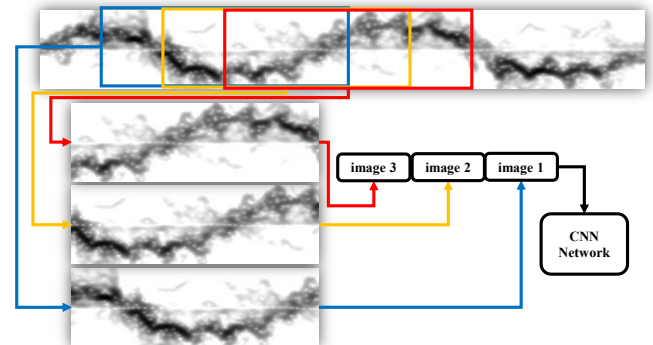


Figure 2 CNN working mode

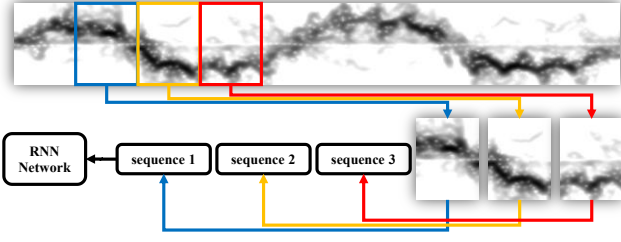


Figure 3 RNN working mode

### 3.3. Overfitting and Data Argumentation

The phenomenon of overfitting is often a problem faced in neural network applications. Because of the powerful learning capabilities of neural networks, when trained with a limited number of training sets, they tend to remember the training set rather than learn the differences between the categories. This phenomenon is usually more severe on small databases than extensive databases.

In this database, a total of about 1754 samples are included; 5 of 6 categories contains around 310 samples, and the last category contains around 200 samples, which is a small amount of data for neural networks. Looking at the easiest standard database in computer vision, MNIST [14] has a total of 10 categories and 7000 samples per category.

Commonly used methods to avoid overfitting, include but are not limited to data augmentation [15], early stopping methods [16], parameter regularization [17] and dropout [18]. In this paper, we selected dropout and data augmentation.

Dropout is a very effective method to prevent overfitting. It randomly discards a part of the output of a layer of the neural network so that the output labels do not over-rely on specific parameters. We added a dropout layer between the two RNN layers.

For the data augmentation, we cut the original samples in the database to increase the number of samples. For example, in the original data set, a sample of the 'walking' category lasts 10 seconds, and by cutting this sample into twenty pieces and each piece lasts 0.5s, we increase twentyfold the number of samples for walking, and because the size of each sample is reduced, we can use a smaller network for classification. However, in actual experiments, we found that if we use samples that are too short of a time-window, such as 0.5s, for training, because of the insufficient amount of information in each sample, the network will not converge, and the final classification performance will decrease. As a result, we decide that when we draw samples from the training set to train the network, we clip the sample to a random length time window, varying from 1 to 4 seconds. In this way, we can ensure that the network training has sufficient convergence speed and have a large number of samples. However, when using the testing set to verify the performance of the network, we use the original samples without clipping.

## 4. Results

In the first experiment, we randomly split the dataset into a training set (90%) and a test set (10%), and use three different methods to process the training set. The first method uses the original data without cutting. The second method is cutting the samples to a fixed 0.5s duration to increase the number of samples. The third is also cutting the samples, but with a random time duration varying from 1 to 4s. And then, neural networks are trained using the training set processed by each method.

The network structure we use is a three-layers LSTM with an input size of 131 and a hidden layer size of 128. Dropout layers with 0.5 probability are added between each two LSTM layers. Dropout layers are used only in the second and third methods. The output of LSTM layers is passed to two fully-connected layers, with size 64 and 6, and finally output through the softmax layer. A synoptic of the model is shown in Figure 4.

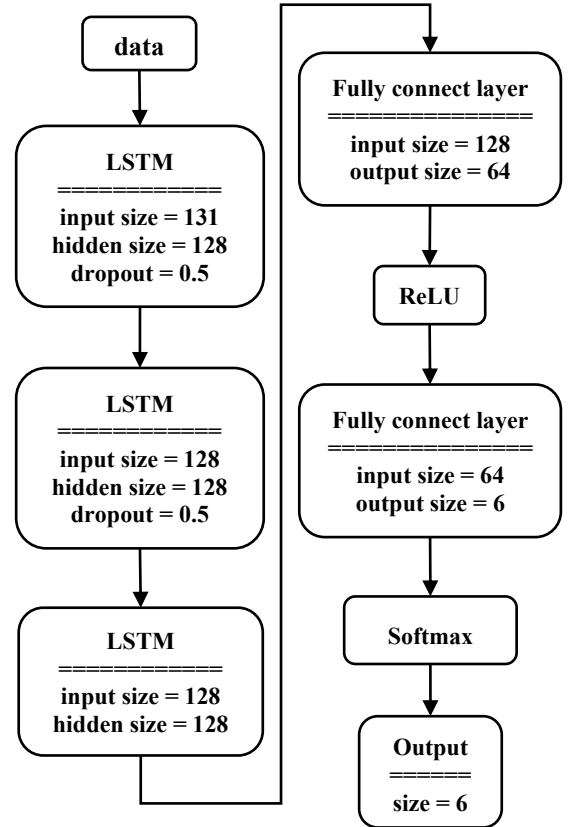


Figure 4 LSTM network for classification

Networks are trained for 1000 epoch using Adam optimizer with a learning rate of 0.001. We recorded the train loss and test loss after each epoch. The result is shown in Figure 5.

The result shows that, without using dropout layers and data argumentation (Method 1), both train and test loss drop very fast, but the test losses increased significantly in the later stage of training, which means that the network is overfitting.

Figure 1 is a line graph showing the training and testing loss of the proposed model across 1000 epochs. The y-axis represents the loss, ranging from 0 to 1.5. The x-axis represents the epoch, ranging from 0 to 1000. The legend indicates six data series: M1 train (solid blue line), M1 test (dotted blue line), M2 train (solid purple line), M2 test (dotted purple line), M3 train (solid orange line), and M3 test (dotted orange line). M1 training loss decreases rapidly from 1.5 to near 0 by epoch 200. M2 training loss decreases from 1.5 to approximately 0.7 by epoch 200 and then fluctuates. M3 training loss decreases from 1.5 to approximately 0.2 by epoch 200 and then fluctuates. Testing losses for all models fluctuate between 0.2 and 1.0 throughout the training process.

In the second experiment, we use the database to train two different RNN networks, LSTM and Gated Recurrent Unit (GRU). The size of the GRU network remains the same as in the first experiment but only replace the LSTM layers with GRU layers. Stratified 10-fold cross-validation is used in the experiment, and in each cross-validation, the training set is processed using the third method in the first experiment. Networks are trained for 500 epochs using Adam optimizer with a learning rate of 0.001. The average testing set accuracy and the standard deviation for two networks are summarized in Table 2. The confusion matrices of both networks are presented in Figure 6 and Figure 7, respectively.

Metrics	GRU	LSTM
Average Testing Accuracy	94.3%	93.9%
Standard Deviation	2.1%	1.5%

Figure 6 Confusion matrix of the GRU network

Figure 7 Confusion matrix of the LSTM network

## 5. Conclusion

We also proposed an innovative data argumentation method for radar signals. By cutting the original samples with random duration, we can significantly increase the number of samples and hence suppress the overfitting phenomenon. The experiment also shows that using a smaller input will speed up the training process.

The authors would like to thank the British Council 515095884 and Campus France – PHC Alliance 44764WK France-UK for their financial support.

## References

- [1] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang and L. Van Gool, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition," in *European Conference on Computer Vision*, Cham, 2016.
- [2] S. Song, C. Lan, J. Xing, W. Zeng and J. Liu, "Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3459-3471, 2018.
- [3] J. Liu, Y. Huang, X. Peng and L. Wang, "Multi-view descriptor mining via codeword net for action recognition," in *IEEE International Conference on Image Processing (ICIP)*, 2015.
- [4] M. G. Amin, Y. D. Zhang, F. Ahmad and K. C. Dominic Ho, "Radar Signal Processing for Elderly Fall Detection: The future for in-home monitoring," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 71-80, 2016.
- [5] Y. Kim and H. Ling, "Human Activity Classification Based on Micro-Doppler Signatures Using a Support Vector Machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328-1337, 2009.
- [6] D. P. Fairchild and R. M. Narayanan, "Classification of human motions using empirical mode decomposition of human micro-Doppler signatures," *IET Radar, Sonar Navigation*, vol. 8, no. 5, pp. 425-434, 2014.
- [7] J. Li, S. L. Phung, F. Tivive and A. Bouzerdoun, "Automatic classification of human motions using Doppler radar," in *International Joint Conference on Neural Networks (IJCNN)*, 2012.
- [8] R. J. Javier and Y. Kim, "Application of Linear Predictive Coding for Human Activity Classification Based on Micro-Doppler Signatures," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1831-1834, 2014.
- [9] J. L. Kernec, F. Fioranelli, D. Chuanwei, Z. Heng, S. Li, H. Hong, J. Lorandel and O. Romain, "Radar Signal Processing for Sensing in Assisted Living: The Challenges Associated With Real-Time Implementation of Emerging Algorithms," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 29-41, 2019.
- [10] S. Z. Gurbuz and G. M. Amin, "Radar-Based Human-Motion Recognition With Deep Learning: Promising Applications for Indoor Monitoring," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16-28, 2019.
- [11] Y. Kim and T. Moon, "Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8-12, 2016.
- [12] B. Jokanović and M. Amin, "Fall Detection Using Deep Learning in Range-Doppler Radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 180-189, 2018.
- [13] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 2015.
- [14] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
- [15] A. Fawzi, H. Samulowitz, D. Turaga and P. Frossard, "Adaptive data augmentation for image classification," in *IEEE International Conference on Image Processing (ICIP)*, Phoenix, 2016.
- [16] R. Gencay and M. Qi, "Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 726-734, 2001.
- [17] B. M. Chirstopher, "Regularization in Neural Networks," in *Pattern Recognition and Machine Learning*, New York, Springer-Verlag, 2006, p. 256.
- [18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.